In the Title:

Please amend the title as follows:

~~Program counter adjustment based on the detection of an instruction prefix~~

Pre-Decoding Bytecode Prefixes Selectively Incrementing Stack Machine Program Counter

In the Specification:

Please enter the following replacement paragraphs:

[0001] This application claims priority to U.S. Provisional Application Serial No. 60/400,391 titled "JSM Protection," filed Jul. 31, 2002, incorporated herein by reference. This application also claims priority to EPO Application No. 03291920.1, filed Jul. 30, 2003 and entitled "Program Counter Adjustment Based On The Detection Of An Instruction Prefix," incorporated herein by reference. This application also may contain subject matter that may relate to the following commonly assigned co-pending applications incorporated herein by reference: "System And Method To Automatically Stack And Unstack Java Local Variables," Ser. No. [[_____]] 10/632,228, filed. Jul. 31, 2003, ~~Attorney Docket No. TI-35422 (1962-05401)~~ now US 7,069,415, granted 06/27/2006; "Memory Management Of Local Variables," Ser. No. [[_____]] 10/632,067, filed Jul. 31, 2003, ~~Attorney Docket No. TI-35423 (1962-05402)~~ now US 7,023,797, granted 04/10/2007; "Memory Management Of Local Variables Upon A Change Of Context," Ser. No. [[_____]] 10/632,076, filed Jul. 31, 2003, ~~Attorney Docket No. TI-35424 (1962-05403)~~ Pub. No. 2004/0,078,522, published 04/22/2004; "A Processor With A Split Stack," Ser. No. [[_____]] 10/632,079, filed Jul. 31, 2003, ~~Attorney Docket No. TI-35425(1962-05404)~~ now US 7,058,765, granted 06/06/2006; "Using IMPDEP2 For System Commands Related To Java Accelerator Hardware," Serial No. 10/632,069, filed Jul. 31, 2003, ~~Attorney Docket No. TI-35426 (1962-05405)~~ now US 7,360,060, granted 04/15/2008; "Test With Immediate And Skip Processor Instruction," Ser. No. [[_____]] 10/632,214, filed Jul. 31, 2003, ~~Attorney Docket No. TI-35427 (1962-05406)~~ Pub. No.

2004/0,024,997, published 02/05/2004; "Test And Skip Processor Instruction Having At Least One Register Operand," Ser. No. [[_____]] 10/632,084, filed Jul. 31, 2003, ~~Attorney Docket No. TI-35248 (1962-05407)~~ Pub. No. 2004/0,153,885, published 08/05/2004; "Synchronizing Stack Storage," Ser. No. [[_____]] 10/632,422, filed Jul. 31, 2003, ~~Attorney Docket No. TI-35429 (1962-05408)~~ now US 7,162,586, granted 01/09/2007; "Methods And Apparatuses For Managing Memory," Ser. No. [[_____]] 10/631,252, filed Jul. 31, 2003, ~~Attorney Docket No. TI-35430 (1962-05409)~~ Pub. No. 2004/0,024,969, published 02/05/2004; "Write Back Policy For Memory," Ser. No. [[_____]]10/631,185, filed Jul. 31, 2003, ~~Attorney Docket No. TI-35431 (1962-05410)~~ Pub. No. 2004/0,024,792, published 02/05/2004; "Methods And Apparatuses For Managing Memory," Ser. No. [[_____]] 10/631,205, filed Jul. 31, 2003, ~~Attorney Docket No. TI-35432 (1962-05411)~~ Pub. No. 2004/0,024,970, published 02/05/2004; "Mixed Stack-Based RISC Processor," Ser. No. [[_____]] 10/631,308, filed Jul. 31, 2003, ~~Attorney Docket No. TI-35433 (1962-05412)~~ Pub. No. 2004/0,024,989, published 02/05/2004; "Processor That Accommodates Multiple Instruction Sets And Multiple Decode Modes," Ser. No. [[_____]] 10/631,246, filed Jul. 31, 2003, ~~Attorney Docket No. TI-35434 (1962-05413)~~ Pub. No. 2004/0,024,990, published 02/05/2004; "System To Dispatch Several Instructions On Available Hardware Resources," Ser. No. [[_____]] 10/631,585, filed Jul. 31, 2003, ~~Attorney Docket No. TI-35444 (1962-05414)~~ now US 7,395,413, granted 07/01/2008; "Micro-Sequence Execution In A Processor," Ser. No. [[_____]] 10/632,216, filed Jul. 31, 2003, ~~Attorney Docket No. TI-35445 (1962-05415)~~ Pub. No. 2004/0,024,999, published 02/05/2004; "Reformat Logic To Translate Between A Virtual Address And A Compressed Physical Address," Ser. No.

[[_____]], 10/632,215 filed Jul. 31, 2003, ~~Attorney Docket No; TI-35460 (1962-05417)~~ now US 7,506,131, granted 03/07/2009; "Synchronization Of Processor States," Ser. No. [[_____]] 10/632,024, filed Jul. 31, 2003, ~~Attorney Docket No. TI-35461 (1962-05418)~~ Pub. No.2004/0,024,988, published 02/05/2004; "Conditional Garbage Based On Monitoring To Improve Real Time Performance," Ser. No. [[_____]] 10/631,195, filed Jul. 31, 2003, ~~Attorney Docket No. TI-35485 (1962-05419)~~ now US 7,392,269, granted 06/24/2008; "Inter-Processor Control," Ser. No. [[_____]] 10/631,120, filed Jul. 31, 2003, ~~Attorney Docket No. TI-35486 (1962-05420)~~ now US 7,434,029, granted 10/07/2008; "Cache Coherency In A Multi-Processor System," Ser. No. [[_____]] 10/632,229, filed Jul. 31, 2003, ~~Attorney Docket No. TI-35637 (1962-05421)~~ now US 6,996,683, granted 02/07/2006; "Concurrent Task Execution In A Multi-Processor, Single Operating System Environment," Ser. No. [[_____]] 10/632,077, filed Jul. 31, 2003, ~~Attorney Docket No. TI-35638 (1962-05422)~~ Pub. No. 2004/0,025,161, published 02/05/2004; and "A Multi-Processor Computing System Having A Java Stack Machine And A RISC-Based Processor," Ser. No. [[_____]] 10/631,939, filed Jul. 31, 2003, ~~Attorney Docket No. TI-35710 (1962-05423)~~ Pub. No. 2004/0,078,550, published 04/22/2004.

[0030] Referring to FIG. 4, a portion of a program may include a plurality of Bytes in register 162 (labeled as Bytes A-G). In Java, those bytes are called Bytecodes. Therefore, for simplicity, these bytes may be referred as Bytecodes in both instruction sets. One or more of those bytes form a Java instruction. The Java instruction set, therefore, may include variable length instructions. In C-ISA, the instruction length may also vary in length (e.g., 2 to. 4 bytes) and like the Java instructions, the length may only be determined

during the decode. In accordance with the preferred embodiments, while the decode logic 152 is decoding Bytecode A, in which Bytecode A may be all or a portion of a current instruction, the next five Bytecodes preferably are provided to pre-decode logic 158. As illustrated, Bytecodes B through F are pre-decoded by the pre-decode logic 158 in parallel with the decoding of Bytecode A. The pre-decode logic 158 preferably determines if any of Bytecodes B through F include a predetermined "prefix." A prefix, as used herein, is a Bytecode that indicates the type of instruction that follows the prefix. For example, a format of an instruction, such as a "wide" instruction, may include a Java wide bytecode used as a prefix. In yet another example, a prefix (e.g., one of the Java implementation dependent and reserved code "Impdepl" Bytecode) may be used as a prefix and may indicate the presence of an instruction that belongs to a particular instruction set. As such, by first determining if an instruction differs in format or if the instruction belongs to a particular instruction set, the decode logic 158 may adjust the decoding behavior for decoding of the instruction.